

LINUX EXPERT

FAQ

Q Sometimes when I try to unmount an NFS partition or eject a CD, I can't because the system says it's being used by another user. Is there an easy way to tell who it is?

A Try the `fusers` command and specify the drive you want to check:

```
# fusers /media/cdrom0
```

This tells you the PIDs of the processes using the resource. Add the `-v` switch to see more detail and the `-u` to add the username. If you think there's a dead process holding on to a resource you want to free up, you can use `fusers` to kill (`-k`) the process too:

```
# fusers -k /media/cdrom0
```

You can also use `fusers` to kill all processes using a directory tree, so you can kick off users who refuse to leave a system about to be shut down:

```
# fusers -k /home
```

It's not particularly nice to be booted off a system, however, so you can soften the blow with the `-i` switch, which asks the user to confirm that the process can die.

Jon Thompson
UNIX/Linux management
and security consultant

linux@computershopper.co.uk

Running Windows programs on Linux

Linux users aren't big Microsoft fans, but there are Windows programs you do need to access under Linux. Jon Thompson gives you a taste of Wine, which lets programs run unchanged

Why can't you run Windows programs on Linux? Both Linux and Microsoft Windows can use the same PC microprocessor. But there are obstacles, not least of which is Windows' use of proprietary libraries. However, developers have spent many years trying to get Windows applications running on Linux. In this month's *Linux Expert*, we show you a technique for running a good selection of Windows XP software on a Linux desktop.

Programs written for one operating system have a number of problems when trying to run on another. Though languages such as ANSI C are portable, the facilities used by compiled programs are not. Different operating systems use different memory layouts, for instance, so they have different resources and ways of doing things. Above all, libraries provided by the operating system, collectively called the system's application programming interface, or API, are bound to be unavailable on other operating systems.

When most programs are compiled, the finished executable does not include all the libraries it needs. This saves space, and a number of programs can access one single shared library rather than each containing duplicate code. Programs that have everything compiled into them are known as statically linked binaries. The rest are described as being dynamically linked; when you run such a program, a piece of code at

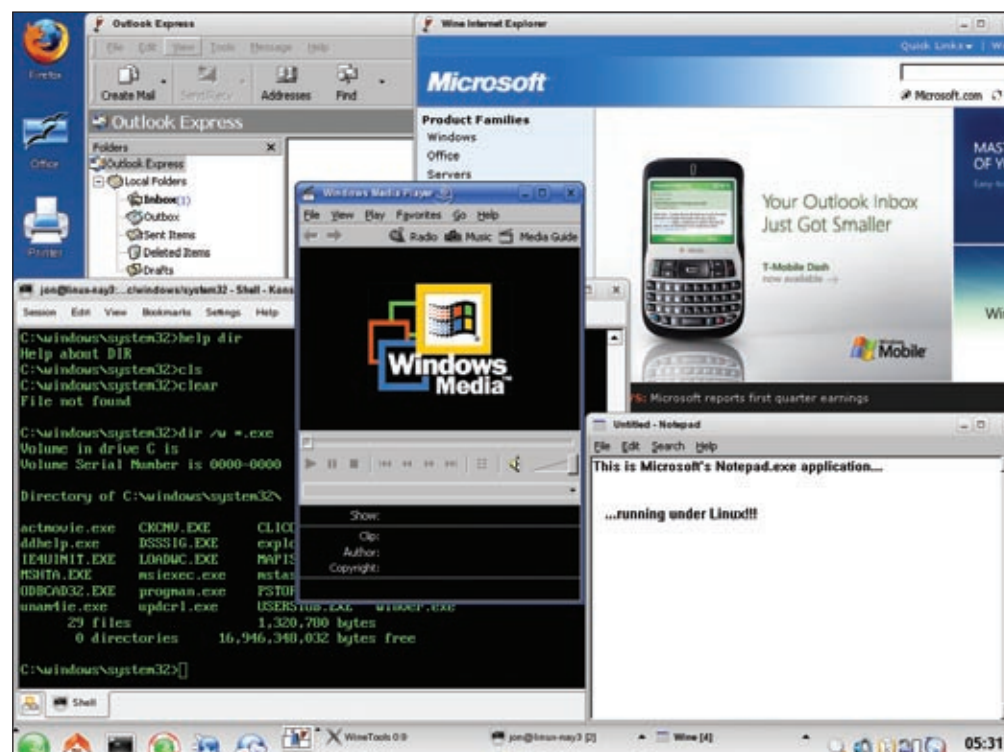
the start loads the libraries the program needs from disk.

Dynamically linked programs have the major advantage that if someone identifies a bug in one of the libraries, or re-implements a library subroutine to make it more efficient, you need only install a new copy of the library rather than recompiling all the applications that use it. In Windows, these libraries are called dynamic link libraries (DLLs). So, if you can re-create in Linux the functionality of the DLLs used by a Windows program, including those for graphical output, in theory the Windows program should be able to run under Linux.

FINE WINE

Wine, which is short for 'Wine is not an emulator', is a project that aims to allow programs compiled under different versions of Windows to run on a Linux system without change. It works by using open-source versions of all the libraries these programs need. The project started in 1993, when Bob Amstadt came up with the idea of reverse-engineering the DLLs used in Windows.

As long as these replacement DLLs exist and have all the required subroutines in the right places, Windows programs shouldn't care whether they are using reverse-engineered libraries or the real thing.



▲ Wine can help you run a wide range of Windows applications under Linux, but some are a little more troublesome than others



But aren't there dozens of large, complex, proprietary libraries to re-implement?

According to Jeremy White of CodeWeavers, which now helps to run the Wine project, about half the libraries found in Windows are never used. This makes the job of getting applications to run a lot easier, as they are probably not reliant on some obscure call to a subroutine in a long-forgotten DLL.

The challenge faced by Wine's approach is immense. Although most of the DLLs needed to run Windows software now have Wine counterparts, the effort that has been expended in writing them has been immense. Not all the function calls in Microsoft's Windows API have proper documentation and some have documentation that is incorrect or is available only to Microsoft staff programmers. Add to this the number of low-level calls and even bugs upon which programs have come to rely (and that the project has to re-implement exactly) and you start to get an idea of how much work is involved.

COMPATIBILITY ISSUES

Rather than acting as a full replacement for a Windows environment, Wine creates a compatibility layer consisting of what the running program thinks of as being the Windows API. This in turn calls the real Windows DLLs. Not only that, but it also creates a Win32-compatible memory layout for the application. To further create the illusion of running under Windows, Wine has exception handling and Microsoft-style threads and processes. The latest versions of Wine also include DirectX and OpenGL support, and even have built-in codecs for Windows MultiMedia (WMM) applications.

Wine can run programs in a few different ways. If there are no DLLs that your target application needs to call, you can simply run it from wherever it is on the PC's Windows disk partition. Wine can also attempt to install Windows programs from their media using the setup.exe program. When you use it to install software in this way, it installs the application on the Linux partition under the directory `~/.wine/c_drive/Program Files`.

Wine uses X11-based graphics to handle the output of Windows programs and this has an interesting side effect. X11 lets the output of an application be sent to another computer, which makes it possible to run processor-hungry Windows applications on a fast central Linux server, with the interface appearing on your terminal, which could be an old, slow Windows or Linux PC running X11 software.

The list of software known to run under Wine grows daily and the Wine project, along with other sources of information, maintains a database of applications, utilities and games guaranteed to run. The public are encouraged to send in details of programs they run under Wine, including a screenshot.

Before we show you some sample Windows programs running under Linux and demonstrate the techniques for getting others to run, it's important to bear in mind that Wine is a hack in the truest sense of the word. It is written and maintained by enthusiasts who are attempting to make software do things it was never designed to do, which is the original definition of a hacker.

Because of that, nothing is guaranteed. New versions of Linux, Wine and the programs you

might want to try installing are released all the time. The WineHQ website (www.winehq.com) has a large database giving details of applications others have managed to install and run, so that should be your first port of call. See the Wine Application Database section on page 185 for more details.

Understandably, Microsoft isn't best pleased with the Wine developers. The Microsoft Update software blocks all attempts from software running under Wine to update itself. Windows Genuine Advantage will also not run under Wine because Microsoft deems it not to be genuine. With these provisos, let's get started by installing Wine itself.

OPENING THE BOTTLE

The simplest way of installing Wine is from the discs on which your Linux distribution came. However, this can become out of date very quickly in the fast-paced world of Linux, and Wine is an active and ever-evolving project. At the time of writing, the current release of Wine is 0.9.23, but that is bound to change by the time you read this. It might also be the case that your distribution doesn't carry loads of extra applications.

Rather than rely on the outdated software in your distribution, it's best to obtain and install the latest release of Wine from www.winehq.org/site/download, especially if you've encountered problems running a Windows program using our current version. This page lists binary and source RPMs for a wide range of mainstream distributions, from Fedora Core to Sun's Solaris 10. Clicking on your distribution will take you to Wine's SourceForge project page. Click on the latest binary release for your distribution and version to download it.

You will also find the third-party WineTools utility on the downloads page. These are designed to walk you through the process of installing troublesome but useful applications if you run into problems. The list of available applications includes Internet Explorer 6, Microsoft Office and Adobe Photoshop. You still need the official distribution media to install these programs, but WineTools will try to make the process easier by automatically setting up the environment and handling the installation process for you.

Before you install Wine, make sure it's not installed already or you could run into problems. If your distribution uses a software-management tool such as SUSE's YaST, you should install Wine using this because it will take care of installing any necessary libraries that are not on your system. For those distributions that don't have an installation tool, or in cases where you previously installed Wine from its original source tarball, go into the directory you unpacked it to and enter the following command as root:

```
# make uninstall
```

If you deleted the unpacked tarball to save space after installing a previous version, you'll have to download and unpack it again with the following command before running make uninstall:

```
# tar xvf <wine tarball>
```

The uninstall command should remove Wine from your system, but not its configuration files. Because of this, you next need to check to make



▲ If you get stuck, try WineTools by Frank Hendriksen, which installs dozens of well-known applications and pieces of system software

sure your home directory doesn't have a file or a directory such as `.wine`. Delete it if it does, and you're ready to install the latest version.

A word of caution: there are purists who prefer to build everything from the source tarballs. While this is commendable, Wine has a number of dependencies in the form of other open-source libraries. Unless you've performed a full Linux distribution installation first, there are bound to be some missing. You'll also need a C compiler such as GNU's gcc, which should be on your distribution media.

HALF FULL OR HALF EMPTY

The documentation says that the best way of checking that your system has everything installed ready for Wine is to watch it fail to install, grab the missing library it complains about and try again until the installation process runs to completion without any errors.

According to the documentation, the Wine installation configuration script can work around some of its minor dependencies by giving slightly less functionality than normal. A full listing of the build output can be found in the `include/config.h.in` file for you to check. Luckily, if you have installed a previous version, and unless there has been a major upgrade to Wine itself, the libraries it depends on should already have been installed.

You can install it as follows:

```
# tar xvf <wine tarball>
# ./configure
# make depend
# make
# make install
```

On the whole, it's far easier simply to install the RPM or update it with a newer version. If you do this manually, you may still receive an error message telling you that required libraries and other software are not installed, but at least the package manager will tell you up front and in one short list, so you can either install the missing libraries from your media or download and install them from the web. To install the Wine RPM on the command line, type this:

```
# rpm -i <wine binary rpm package>
```

To update a previous installation with a newer version, enter:

```
# rpm -U <wine binary rpm package>
```

TESTING, TESTING

Once you've installed Wine, try running a program. We'll start with a basic one, Windows Notepad. Open a terminal and enter:

```
# wine "c:\windows\notepad.exe"
```

Notepad should appear. It's a very old and well-behaved program that's guaranteed to run in Wine, and it's a good indicator that Wine is installed properly.

Go to the directory `.wine/drive_c` and look at its contents. Both the Windows 'Program Files' and 'Windows' directories are there, among others. This is Wine's fake C: drive that makes applications believe they're running under Windows. Under the Windows subdirectory you'll find the usual system and system32 directories containing the system's DLLs.

When navigating this directory tree, you'll need to escape the spaces in the directory names such as 'Program Files' for Linux to recognise them properly. Do this as follows:

```
# cd Program\ Files
```

Next, let's run the Wine configuration program:

```
# winecfg
```

Winecfg runs through several system checks before presenting a graphical configuration tool consisting of a number of tabbed windows. Let's begin with a simple demonstration of configuring Wine to run programs without having to enter the entire path to them.

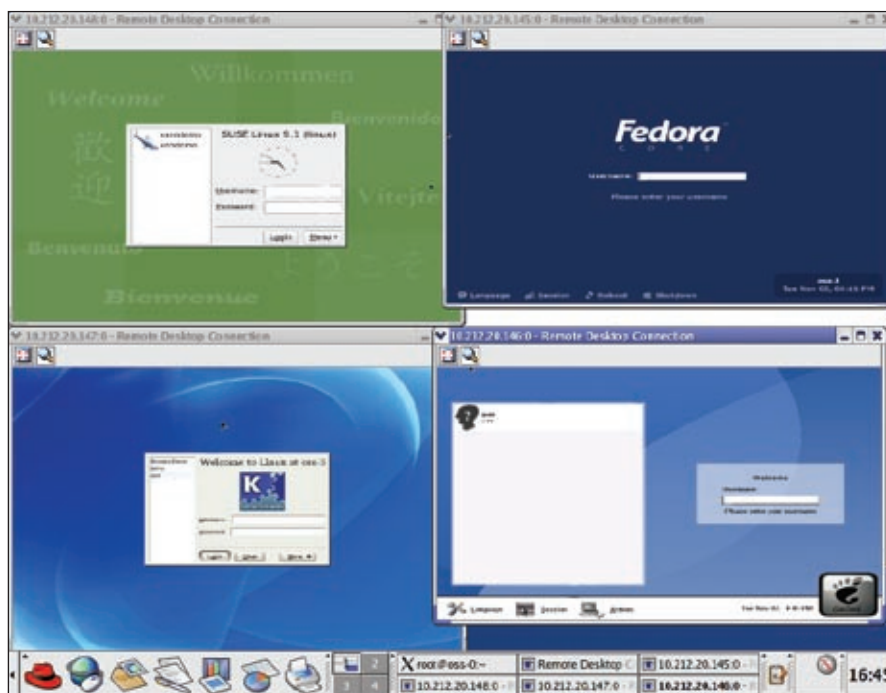
Click on the Applications tab and press the Add Application... button. A file selector opens showing the `drive_c` directory. Double-click the Windows icon and the notepad.exe icon from the list that appears. Click OK to dismiss winecfg and enter the program's name on the command line. In Wine, you don't need the .exe extension:

```
# wine notepad
```

In Windows, most application settings are stored in a central repository called the Registry. Wine also uses a registry, which is stored in the `.wine` directory in your home directory. On Linux this is stored in a set of plain text files that can be edited directly, unlike its Windows counterpart. The Windows applications you install in Linux use this to store their settings, so don't expect to see any changes you make being replicated when you boot back into Windows.

The Registry is organised into various trees or 'keys'. There are many of these and some are obscure. The Wine version of the Registry is stored in the following files:

FILE	KEYS CONTAINED	DESCRIPTION
System.reg	HKey_Local_Machine	Everything about the Windows system (faked by Wine)
User.reg	HKey_Current_User	Configuration data for different users
Userdef.reg	HKey_Users\default	Default user information



▲ The open-source Xen virtualisation software allows you to run multiple operating systems including Windows

If you're experienced at using the Windows RegEdit utility, you'll be pleased to know that you can find it under `.wine/drive_c/windows`. You can run it directly with the following command:

```
# wine ~/.wine/drive_c/windows/regedit
```

WINETOOLS TO THE RESCUE

If you have the distribution media for Microsoft Office, you can attempt to install it under Linux. Simply navigate to the CD and type:

```
# wine SETUP.EXE
```

The usual Office setup program runs and guides you through the process. You have to enter your licence key. You can try installing other Windows applications in this way, too. As noted earlier, Wine is hacking in the truest sense of the word. It pays to read the Wine documentation right the way through before you start. It's also worth searching the internet to see if anyone has already solved the problems you may encounter during installation.

This is because it can be frustrating trying to get software running under Wine when you're not sure where to start and scratching around for information. Luckily, several people have been good enough to release tools that, provided you have the original media, you can use to install Windows programs in the Linux partition.

One such tool is WineTools, by Frank Hendriksen. This utility will take much of the sting out of installing programs the author has found to work well under Wine, and also handle all the Wine configuration options. You can download WineTools from the download page at the Wine website, or from the author's own site at www.von-thadden.de/joachim/WineTools. Download the latest RPM and install it with the following command:

```
# rpm -i <winetools rpm>
```

You can now run it by typing:

```
# winetools
```

When you first run WineTools, after a few disclaimers, it will ask you for your username, organisation and the location of your CD or DVD drive. Don't worry if you don't know this last item because you can specify the path to the setup program for an application later.

The main menu presents you with some options covering base setup, the installation of Windows system software (which mainly involves runtime libraries) and software that has been tested and is known to run with Wine 0.9. It also allows you to edit Linux's fake registry, manage running processes and play with the Wine configuration.

First of all, click Base setup, then press the OK button. The first task is to create a fake Windows drive. Select the option and press OK. WineTools will download and install DOM98, so go back into this menu and select the appropriate option before pressing OK to install it. WineTools will handle the download automatically. Do the same with the TrueType Font Arial option and Internet Explorer 6.0 SP1 English. A lot of other Microsoft products rely on IE being installed, and it's a good test to make sure everything is working, too.

WineTools should have created a command in your local binaries directory (`~/bin`) that will run IE6 if you enter the following:

```
# ie6 <url>
```

Once running, Wine will interrupt and explain that it needs to download the Mozilla Gecko extension to render HTML. This takes a couple of minutes, after which you should see the contents of the site you specified with the ie6 command.

If you have an official copy of Microsoft Office, you can try installing it using WineTools. Put disc 1 into the CD drive, go to the 'Software known to work with Wine' option and press OK.



You'll see a long list of collapsed trees, each covering a different type of application. Expand the Office tree, select your version and press OK. You'll be asked if you want to use the default version. Press Cancel and a file selector dialog will pop up. Navigate to your Office media and select setup.exe. Then press OK.

If you've tried unsuccessfully to install Office before, a dialog will pop up asking if you want to continue. Press Yes and the Office installation wizard will appear. You'll need to enter your details as you normally would, and you'll need your licence key to complete the installation.

After a few minutes, WineTools will pop up a final dialog box explaining that it has placed several handy commands in your personal bin directory so you can run the various applications directly. Unfortunately, under SUSE 10.1, none of the filenames was correct. If you use this distribution, go into the `~/wine/c/Program Files/Microsoft Office/Office` directory and run the application of your choice from there:

```
# wine <application>#
```

The list of software WineTools will install is extensive. If you're installing free software, such as Google Earth, the program will even download the executable for you from the web.

TROUBLESHOOTING

As it's an unofficial piece of software, there are bound to be problems when installing certain applications under Wine, but luckily help is at hand. Frank Hendriksen has a very useful website about using Wine, which you can find at www.frankscomer.org. Hendriksen has created and made available a comprehensive list of installation instructions for a wide range of products. Help is at hand from Wine itself, too.

Wine comes with a number of useful utilities that can help if you get in a muddle. The first is wineboot. Enter this command, and any Windows software installation procedure that requires a reboot will believe one has just

occurred. Microsoft Office insists you reboot your computer after installing it, for instance, and wineboot will fool it nicely. Another very useful program is called uninstaller. Run it and it displays a list of Windows software installed under Linux. Uninstaller will also run the uninstall scripts for your applications.

As it's a highly unofficial method of making Windows applications run under Linux, Wine is always going to be somewhat hit-or-miss in its ability to run individual applications. Though the developers have had to piece together and reverse-engineer the content of a large number of proprietary libraries, Wine does a hard job increasingly well. It won't equal the level of compatibility provided by a virtual machine running the native operating system but, if there is a Windows application you just can't live without, it is worth trying before investing in a virtualisation product.

WINE APPLICATION DATABASE

Before trying to get a Windows application to run under Linux using Wine, it's a good idea to consult the online Wine Application Database at <http://appdb.winehq.org>. The database contains several lists of applications. The platinum list contains programs known to install and run perfectly under Wine without any modification. The much larger gold list contains programs that install and run using Wine, but which require some minor tweaks to the Wine environment, too.

The silver list contains software that, while it runs OK, has some minor issues. The Bronze list, as you may expect, contains titles that have some known issues. Finally, there's the garbage list, containing programs that simply won't play.

You can use the Wine AppDB website in a number of ways. First, it's there for you to see if someone else has discovered whether the application you want to install will run. It's also there to share your knowledge with others by creating an account and adding to it. More importantly, the database contains

documentation and how-to guides that help you get the more troublesome applications to run on your Linux system.

VIRTUAL ALTERNATIVES

Wine looks as if it will remain a work in progress for some time. It's becoming more capable with every release, and many programs written for Microsoft Windows programs now install and run under Linux without any trouble because of it. Equally, however, many don't. For users who need a guarantee that programs will run perfectly, though, there are alternatives.

Virtual machines – an emulation of the computer's hardware entirely in software – allow the user to boot a computer with one operating system, then, from within the running environment, run a second operating system as if it were running on a completely different computer. The second operating system believes it is running normally on a regular PC. Depending on how much disk space and RAM are available, and how fast the processor runs, you can boot several operating systems or even multiple copies of the same one.

There are various techniques for creating virtual machines. The most basic is simply to emulate the processor itself. Each instruction in the program causes the virtual machine to call a corresponding subroutine. This technique enables one machine to simulate the processor perfectly from another but, being a straight simulation, it isn't very efficient.

Another method is to perform dynamic recompilation as the operating system and applications run. This might seem like more work, but it increases the efficiency of the virtual machine dramatically because it can enable the software to use single instructions that the original program expresses as a long list of code. The host processor might, for instance, have an instruction to copy a block of data from one place to another. If the program running on the virtual machine contains a long list of basic instructions instead, it can replace this with a subroutine that runs its more efficient native counterpart. This is the method preferred by Microsoft's Virtual PC product.

A third, even more efficient approach, is to run the code directly, wherever possible, and to use dynamic recompilation elsewhere. One well-established commercial product that uses this third technique is VMware, of which there are several versions. These range from the personal Workstation version, designed for desktop use, to the enterprise-class ESX Server that is designed to boot in place of an operating system and then have the operating systems boot from within it.

The open-source community is also showing great interest in virtual machines. There is one that currently ships with many mainstream Linux distributions. The open-source Xen virtualisation software from the University of Cambridge usually comes with an alternative kernel configured with the Xen extensions that gives an extra Linux boot option. Xen, like Wine, is very much a work in progress, but it looks as if it will give VMware a run for its money within a year or two. 



▲ VMware allows you to run Windows XP under Linux, thereby granting you access to its applications even if Wine won't install them

Next month

CONTROL LINUX STARTUP

Using Inittab to customise how Linux starts